Bearer Token Sicherheit

mit ASP.NET Core

Thorsten Kansy (tkansy@dotnetconsulting.eu)

Meine Person- Thorsten Kansy

Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor

Microsoft SQL Server

.NET Core

ASP.NET Core

Azure Cosmos DB

DOTNETCONSULTING

Mein Service- Ihr Benefit

- Individuelle Inhouse Trainings
- (Online on-demand) Projektbegleitung
- Beratung
  - Problemanalyse und Lösungen
  - Technologieentscheidungen

DOTNETCONSULTING

Agenda

- Grundlagen
  - Was ist Bearer Token Security
  - Was ist JWT?
- Claims
- Token
  - Erstellen
  - Übermitteln
  - Prüfen
- Authorization
- API-Key

Verwendete Software

Software-Versionen

- Entwicklungsumgebungen
  - Visual Studio 2022 (17.4.0+)
  - Visual Studio Code
  - JetBrains Rider
  - …
- .NET (Core)
  - .NET 5.0+

Grundlagen

Grundlagen- die Ausweispapiere bitte

- “Give access to the bearer of this token.”
- Jeder Request liefert einen Bearer, ein Token mit
  - „Normalerweise“ via Header `Authorization: Bearer <Token>`
  - Wird normalerweise von einem Server bei der Anmeldung erzeugt
  - JWT (Json Web Token) wird oft verwendet, nehmen wir auch

`https://swagger.io/docs/specification/authentication/bearer-authentication/`

Was sollte im “Token” stehen? Was nicht?

- Unveränderliche Informationen
  - Name
  - EmailAdresse
  - (Relevante)Rollen-/ Gruppenzugehörigkeiten

- Veränderliche Informationen
  - Alter
- Blobs
  - Foto
- Sensibles
  - Kennwörter
  - PINs

Sicherheitsmerkmale

- Herausgeber
  - Issuer
- Signatur des Herausgebers
  - IssuerSigningKey
- Gültigkeit
  - LifeTime
- …

Struktur des Ausweises - Schema

- JWT Json Web Token
  - Base64 kodiertes JSON

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1laWQiOiJ0a2FucyIsImdpdmVuX25hbWUiOiJUaG9yc3RlbiIsInVuaXF1ZV9
uYW1lIjoiS2Fuc3kiLCJlbWFpbCI6InRrYW5zeUBkb3RuZXRjb25zdWx0aW5nLmV1IiwiaHR0cDovL3NjaGVtYXMueG1sc29hcC5vcmc
vd3MvMjAwNS8wNS9pZGVudGl0eS9jbGFpbXMvc2lkIjoiZGVtNVEwZVkiLCJyb2xlIjpbIlJvbGVBIiwiUm9sZUIiLCJSb2xlRGV0YWl
scyIsIlJvbGVDb250YWN0cyIsIlJvbGVUYXNrcyIsIlJvbGVEb2N1bWVudHMiLCJSb2xlU2VjdXJpdHkiLCJSb2xlRGVsZXRlIl0sImh
0dHA6Ly9zY2hlbWFzLmRvdG5ldGNvbnN1bHRpbmcuZXUvd3MvMjAyMS8wMy9pZGVudGl0eS9jbGFpbXMvcG9saWN5IjoiNiIsImh0dHA
6Ly9zY2hlbWFzLmRvdG5ldGNvbnN1bHRpbmcuZXUvd3MvMjAyMS8wMy9pZGVudGl0eS9jbGFpbXMvY3VsdHVyZSI6ImRlLURFIiwibmJ
mIjoxNjYyNzEzNTA0LCJleHAiOjE2NjMzMTgzMDQsImlhdCI6MTY2MjcxMzUwNH0.2VnHoVtbo6Vbr5Q9fKbuBifPUVc8Y84g9GgSYWS
EGGw

(732 Bytes)
```

https://jwt.io

DOTNETCONSULTING

Demo

Claims

Einträge in dem Ausweis - Claims

Standards unter `System.Security.Claims.ClaimTypes`

```
namespace System.Security.Claims
{
    public static class ClaimTypes
    {
        public const string Actor = "http://schemas.xmlsoap.org/ws/2009/09/identity/claims/actor";
        public const string PostalCode = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode";
        public const string PrimaryGroupSid = "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarygroupsid";
        public const string PrimarySid = "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid";
        public const string Role = "http://schemas.microsoft.com/ws/2008/06/identity/claims/role";
        public const string Rsa = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/rsa";
        public const string SerialNumber = "http://schemas.microsoft.com/ws/2008/06/identity/claims/serialnumber";
        public const string Sid = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/sid";
        public const string Spn = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/spn";
        public const string StateOrProvince = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/stateorprovince";
        public const string StreetAddress = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/streetaddress";
        public const string Surname = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname";
        public const string System = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/system";
        public const string Thumbprint = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/thumbprint";
        public const string Upn = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn";
        public const string Uri = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/uri";
        public const string UserData = "http://schemas.microsoft.com/ws/2008/06/identity/claims/userdata";
        public const string Version = "http://schemas.microsoft.com/ws/2008/06/identity/claims/version";
        public const string Webpage = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/webpage";
        public const string WindowsAccountName = "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname";
        public const string WindowsDeviceClaim = "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsdeviceclaim";
        public const string WindowsDeviceGroup = "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsdevicegroup";
        public const string WindowsFqbnVersion = "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsfqbnversion";
        public const string WindowsSubAuthority = "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowssubauthority";
        public const string OtherPhone = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/otherphone";
        public const string NameIdentifier = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier";
        //
        // Summary:
        //     The URI for a claim that specifies the name of an entity, http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name.
        public const string Name = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name";
```

Einträge in dem Ausweis - Claims

Eigene Claims möglich

```
public static class JwtCustomClaims
{
    public const string Culture =
                  "http://schemas.dotnetconsulting.eu/ws/2021/03/identity/claims/culture";
    public const string TransactionId =
                  "http://schemas.dotnetconsulting.eu/ws/2021/03/identity/claims/transid";
    public const string Policy =
                  "http://schemas.dotnetconsulting.eu/ws/2021/03/identity/claims/policy";
}
```

Einträge in dem Ausweis - Claims

Claims können mehrfach vorkommen

```
public static string? UserId(this ClaimsPrincipal ClaimsPrincipal)
{
    return ClaimsPrincipal?.Claims?
       .FirstOrDefault(w => w.Type! == ClaimTypes.NameIdentifier)?.Value;
}
```

Demo

Token

Token erstellen

`Microsoft.IdentityModel.Tokens.SecurityTokenDescriptor`

```
SecurityTokenDescriptor tokenDescriptor = new()
{
    Subject = new ClaimsIdentity(new Claim[]
    {
            // Claims
            new Claim(ClaimTypes.Email, "tkansy@dotnetconsulting.eu"),
            // Custom types
            new Claim(JwtCustomClaims.Culture, "de-DE")
    }),

    Expires = DateTime.UtcNow.AddSeconds(3600),
    SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha256Signature)
};
```

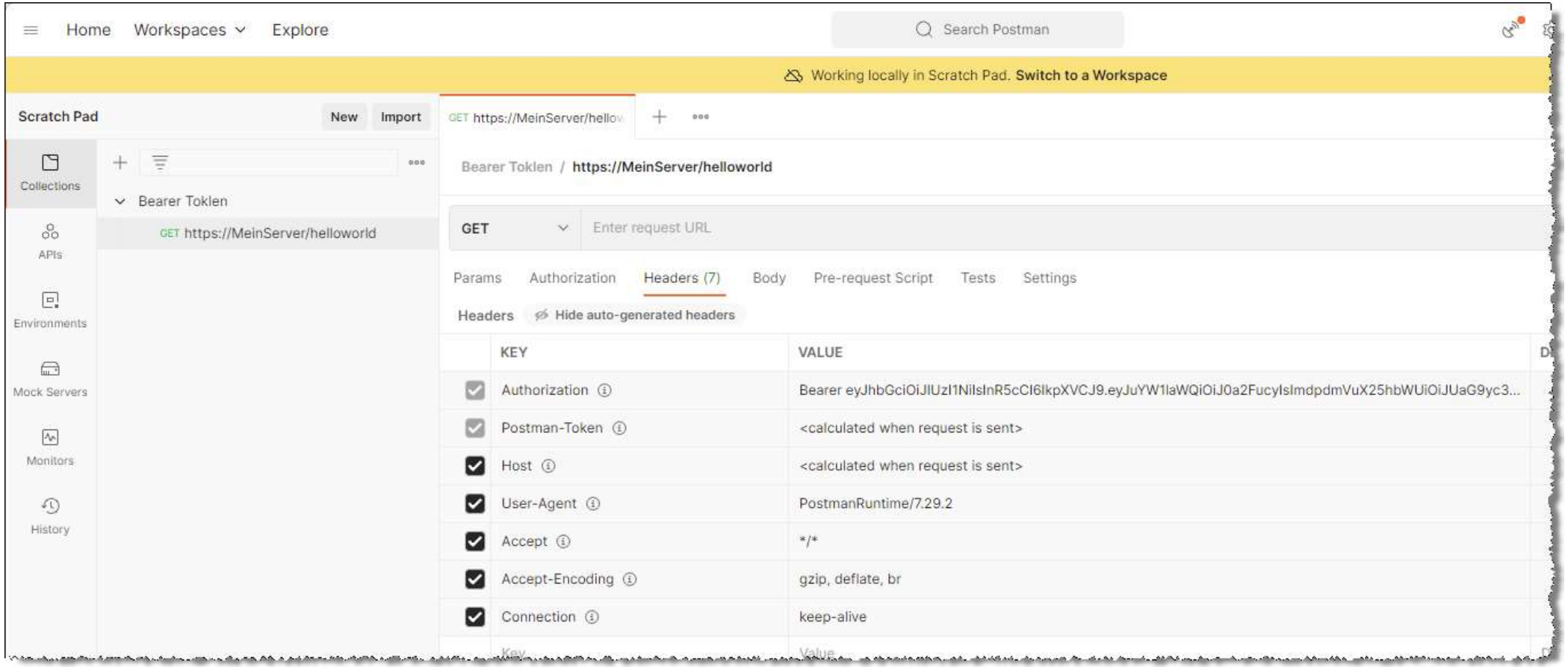
Demo

Token übermitteln/ speichern

Freie Wahl der Übermittlung/ Bereitstellung

- Header
- Body
- Code/ Datei
- …

Authorization Header



DOTNETCONSULTING

Custom Cookie

`Microsoft.AspNetCore.Authentication.JwtBearer. JwtBearerEvents`

```
options.Events = new JwtBearerEvents()
{
    OnMessageReceived = context =>
    {
        string tokenKey = context.Request.Query["t"];
        if (tokenKey is null)
            context.Fail(new JwtValidationException());
        context.Token = context.Request.Cookies[$"JwtToken-{tokenKey}"];

        // Oder fest codiert?
        context.Token = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. ";
        return Task.CompletedTask;
    }
};
```

Demo

Token validieren

`Microsoft.AspNetCore.Authentication.JwtBearer.JwtBearerOptions`

```
...
options.RequireHttpsMetadata = false;
options.SaveToken = true;
options.TokenValidationParameters = new()
{
    ValidateIssuerSigningKey = true,
    IssuerSigningKey = new SymmetricSecurityKey(key),
    ValidateIssuer = false,
    ValidateAudience = false,
    ValidateLifetime = true
};
...
```

Demo

Roles+Authorization

Roles + Authorize-Attribute

ClaimTypes.Role

```
// Rollen
new Claim(ClaimTypes.Role, "RoleA"),
new Claim(ClaimTypes.Role, "RoleB"),
new Claim(ClaimTypes.Role, "RoleC"),
```

Authorize-Attribute

```
[Authorize(Roles = "RoleB,RoleC")]  // (RoleB oder RoleC)
[Authorize(Roles = "RoleA")] // und RoleA
```

DOTNET CONSULTING

Demo

API-Key

Kombination mit anderen Sicherheits-Schemata

z.B. API-Key

```
// Add services to the container.
builder.Services.AddControllers(o =>
{
    if (apiKeySettings.ProtectWithApiKey)
        o.Filters.Add(new ApiKeyFilter(apiKeySettings));
});
```

```
public void OnAuthorization(AuthorizationFilterContext context)
{
    ...
    // Verify API key
    string apiKey = context.HttpContext.Request.Headers[APIKEYNAME].ToString();

    if (string.Compare(_apiKey, apiKey) != 0)
        context.Result = new UnauthorizedResult();
}
```

DOTNETCONSULTING

Demo

Fragen? Jetzt oder später!

Kontakt

E-Mail
tkansy@dotnetconsulting.eu

Telefon
+49 (0) 6187 / 2009090

Microsoft Teams
Meet now

LinkedIn
Link me

XING
Xing me

X (Twitter)
@tkansy

DOTNETCONSULTING

Bewertung der Session

DOTNET CONSULTING

www.dotnetconsulting.eu

SQL Server meets .NET (Core)- professionally!

Ich berate, coache und trainiere im Bereich Entwicklung von .NET (Core) Anwendungen mit Microsoft SQL Server- mit Allem, was dazu gehört- und was man vielleicht weglassen sollte.

DOTNETCONSULTING